

# Parallel VLSI Equalizer Architectures for Multi-Gbps Satellite Communications

Andrew A. Gray

Jet Propulsion Laboratory/CalTech  
4800 Oak Grove Drive, Pasadena, CA 91109

Scott D. Hoy, Parminder Ghuman

NASA/Goddard Space Flight Center  
Greenbelt Road, Greenbelt, MD 20771

**Abstract**—This paper provides an overview of a new very large scale integration (VLSI) architecture for implementing a frequency domain least-mean squares (LMS) complex equalizer [1,2]. The architecture incorporates a simple *sub-convolution* method, digital vector processing, specialized FFT-IFFT hardware architectures, and the discrete Fourier transform-inverse discrete Fourier transform (DFT-IDFT) overlap and save filter method [3]. A key property of the new architecture is that the equalizer tap length may be chosen completely independently of the FFT-IFFT lengths and input data block lengths. Theoretically unlimited tap lengths are possible with short FFT-IFFT pairs. It will be demonstrated that the new parallel architecture is very well suited for processing multi-Gbps digital communication data rates with relatively low speed CMOS hardware. The VLSI equalizer architecture presented processes complex demodulated symbols at  $1/4^{\text{th}}$  the symbol rate. The parallel equalizer, operating on one sample per symbol, has 32 coefficients, is decision directed, and will process data modulated with quadrature phase-shift keying (QPSK) and 16 quadrature amplitude modulation (QAM). The equalizer will be integrated into the 2.4 Gbps all-digital wireless parallel demodulator application specific integrated circuit (ASIC). The receiver is currently being development by JPL/CalTech and NASA's Goddard Space Flight Center. This parallel all-digital receiver designed for satellite communications operates at  $1/16^{\text{th}}$  the analog-to-digital sample rate. Finally, a complexity comparison between this equalizer architecture and the traditional frequency domain fast LMS equalizer is given.

## I. Introduction

The only true limitation to the data rates an all-digital receiver can process is the analog-to-digital converter. Digital receivers implemented in complementary metal oxide semiconductor (CMOS) hardware generally have substantially lower clock rates than the fastest commercially available analog-to-digital converters (A/Ds). Therefore, CMOS digital receivers and equalizers using traditional serial algorithms for digital communications process data rates approximately 15-20 times lower than the *maximum Nyquist data rate* possible with these A/Ds. Within NASA earth science program data rates in excess of 1 Giga-bits per second (Gbps) are planned in the next 3 years, with even higher data rates to follow [4].

The research described in this publication was carried out as part of a task funded by the *Technology and Applications Program* (TAP) at the Jet Propulsion Laboratory, California Institute of Technology, under contract with the National Aeronautics and Space Administration.

Existing earth orbiting satellites support data rates of several hundred Mega-bits per second (Mbps). NASA's next-generation Telecommunication and Data Relay Satellite System (TDRSS) will support data rates up to 800 Mbps. In addition numerous modulation types are available and varying data rates are required, making great flexibility in receivers necessary.

Advances in GaAs have made processing rates of several Giga-hertz (GHz) possible. However, the widespread use of high-speed GaAs components is costly in terms of both nonrecurring engineering costs and reproduction costs. It is difficult, if not impossible, to implement all of the functionality of all-digital receivers for modern satellite communications on a single GaAs application specific integrated circuit (ASIC). CMOS has much higher transistor density and typically lower nonrecurring engineering and reproduction costs than GaAs. These factors have lead NASA and JPL to developm *the all-digital parallel receiver* for processing near-maximum Nyquist data rates [5,6,7,8,9]. NASA and JPL are currently developing the next generation parallel digital receiver incorporating a parallel equalizer presented here. A high-level block diagram of the receiver is given in Figure 1.

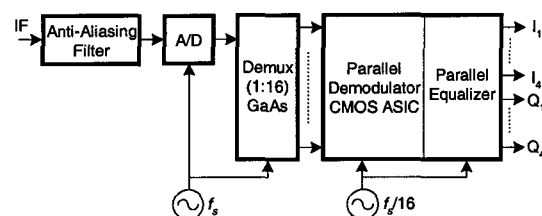


Figure 1. Parallel Digital Receiver

The objective of the parallel algorithms and architectures is to remove the limitation caused by the processing rate difference of GaAs and CMOS as illustrated in Figure 1, with the CMOS demodulator ASIC operating at  $1/16^{\text{th}}$  the sample rate. The equalizer developed here requires  $1/4^{\text{th}}$  rate parallel processing since it operates on one sample per symbol; the demodulator processes 4 symbols per demodulator clock cycle, or 16 samples per clock since 4 samples per symbol are assumed.

## II. Parallel LMS Equalizer and Very High Rate Processing

Figure 2 illustrates a  $1/4^{\text{th}}$  rate frequency domain fast LMS algorithm. This is a specific manifestation of the algorithm

given by Haykin [1] and Shynk [10]. Here  $x(n)$  is the sequence of the complex demodulated baseband QPSK or 16-QAM symbols,  $y(n)$  is the equalizer output and  $d(n)$  is the desired response created from  $y(n)$  (decision directed equalizer [2,11]). We have chosen the FFT-IFFT lengths such that we may achieve the desired rate reduction for hardware implementation. However, the maximum number of coefficients is 5 or less; the architecture in Figure 2 actually implements 4 coefficients. Obviously larger FFT-IFFT pairs may be used to achieve larger tap length; however, this leads to a more complex design and more transistors in a concurrent hardware implementation. The algorithm is typically more computationally efficient or “faster” than the traditional time domain approaches for many applications.

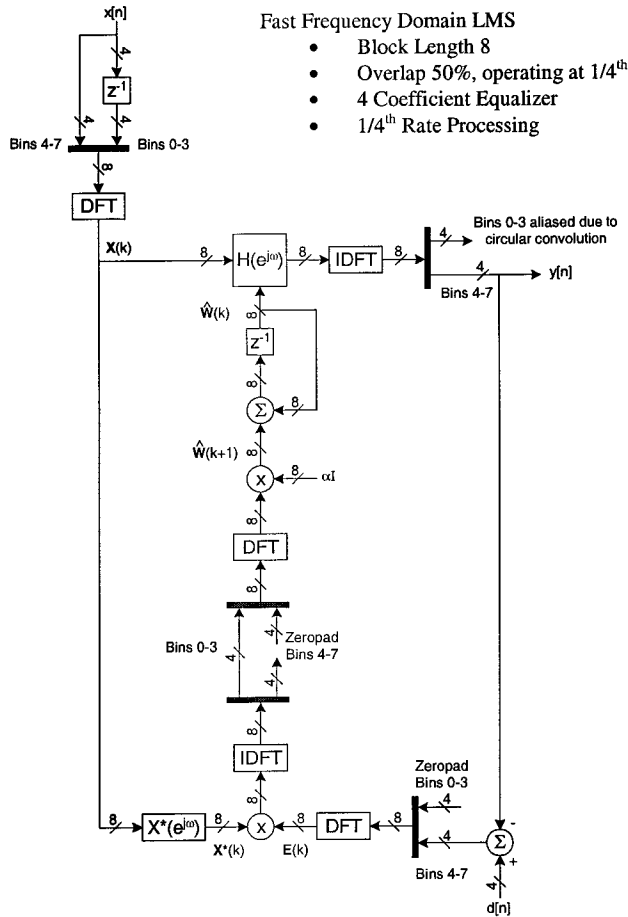


Figure 2. Frequency Domain Fast LMS [1,10]

However, if a very high rate hardware implementation of the algorithm is the goal, computational efficiency is not necessarily the primary design consideration. The architecture may be tremendously computationally efficient, but it may require so many transistors for concurrent implementation it may not be possible to implement.

Alternatively, if hardware reuse is used – an extreme example of this is a pure software implementation – with maximum clock rate of  $\text{clk}_{\text{Max}}$ , the architecture is not capable of processing as much data as the full parallel or concurrent architecture operating at  $\text{clk}_{\text{Max}}$ . A hardware design employing partial concurrent or parallel operation and partial hardware reuse is often highly specialized to a specific algorithm and is most often tedious to design and results in a design difficult to debug and test.

The new equalizer architecture developed here allows trades to be made between computational efficiency, and hardware complexity and processing rate. The trade between computational efficiency and the rate and complexity of hardware is made by eliminating the lower bound on FFT-IFFT lengths of Figure 2 created by the equalizer tap length. Through the methods developed here using *sub-convolution*, this lower bound is completely eliminated through simple and generic digital signal processing methods. The FFT-IFFT lengths may be chosen completely independently of the equalizer tap length, and in fact will be determined by the processing rate reduction desired.

The goal is to create an equalizer architecture that operates in parallel without hardware reuse for the system illustrated in Figure 1. The FFT-IFFT lengths are chosen to give the desired decrease in processing rate, in this case 4 resulting in FFT-IFFT length of 8. The system of Figure 1 has 4 samples per symbol, the demodulator operates at  $1/16^{\text{th}}$  the A/D rate,  $1/4^{\text{th}}$  the symbol rate, and the non-fractionally spaced equalizer operates at  $1/4^{\text{th}}$  the symbol rate [6]. The coefficient length of the equalizer is chosen to be 32 to meet system requirements. Traditional methods do not allow a 32-coefficient equalizer to be implemented with 8-point FFT-IFFT pairs. We now show this can be readily achieved with the simple *sub-convolution* method.

### III. Parallel Sub-Convolution Filter Banks

Figure 3 illustrates a parallel DFT-IDFT filtering architecture for frequency domain filtering or correlation using the overlap and save method. The DFT-IDFT length is  $L+1$  ( $L$  is odd), and  $M$ , the downsample rate, is the number of samples the input window “slides”. The architecture in Figure 3 has 50% input vector overlap, that is the downsample rate, is equal to half the input vector length,  $M = (L+1)/2$ . With such a architecture a  $M+1$  tap filter may be implemented in the frequency domain.

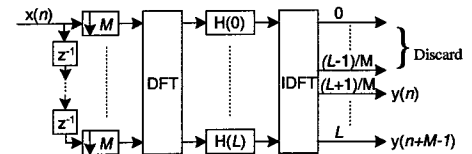


Figure 3. Overlap-and-Save FIR Filter

The filter,  $h(n)$ , is zero-padded to length  $L+1$  and then transformed to the discrete frequency domain via the DFT, to

obtain the frequency domain coefficients,  $H(k)=\text{DFT}\{h(n)\}$  [3]. It is obvious that any FIR filter with an order  $M$  or less can be used with this same architecture. Similar derivations for FIR filters have been developed for implementation in software and hardware. The limitation to all of these methods in a concurrent VLSI implementation is that the DFT-IDFT, or fast Fourier transform-inverse fast Fourier transform (FFT-IFFT) lengths are increased to increase the order of the FIR filter to be implemented.

Consider the simple convolution sum of equation 1. The convolution may be broken into numerous sub-convolutions, each time shifted input convolved with a *sub-filter*, as indicated.

$$y(n) = \sum_{k=0}^{L+1} x(n-k)h(k) = \sum_{k=0}^{j_1} x(n-k)h(k) + \sum_{k=j_1+1}^{j_2} x(n-k)h(k) + \dots + \sum_{k=j_R}^{L+1} x(n-k)h(k) \quad (1)$$

First we observe that each sample vector input to the DFT of Figure 3, and therefore the frequency domain vector, is a time delay of  $M$  samples from the next sample vector input. From (1), it is obvious that each of the sums are themselves a convolution with a block of the filter or *sub-filter*, we call these *sub-convolutions* with sub-filters, the sum of their outputs is equal to the convolution of the input,  $x(n)$ , with the filter  $h(n)$ . Each of these sub-convolutions may be implemented in the frequency domain using the technique illustrated in Figure 3, then the results summed to yield the convolution output. To break a convolution up into  $R$  equal length sub-convolutions, each  $(L+1)$  in length, using this method would require  $R$  DFTs,  $R$  IDFTs, and  $R$  sub-filters. Assuming 50% overlap, the DFT-IDFT pairs would each be in  $(L+1)$  length, however simplifications requiring only one DFT-IDFT pair are possible with one additional constraint. We can derive the constraint simply by realizing that each input vector to the DFT of Figure 3 is a shift in time from the previous or next vector by  $M$  samples, therefore each frequency domain vector is separated in time from the previous or next vector by  $M$  sample periods. From (1), if  $j_i + M = j_{i+1} \forall i$ , that is the time delay between each sub-filter is equal to the time delay between time-consecutive input vectors (figure 3), then the convolution of (1) may be calculated in the frequency domain by simply delaying the frequency domain vectors and multiplying by the appropriate frequency domain sub-filter. These sub-filters are generated as follows.

$$H_k(i) = \text{DFT}\{h_k(n)\} \quad i=0, \dots, L, \quad k=1, \dots, R \quad (2)$$

$$n=0, \dots, L$$

and  $h_k(n)$  is the  $k^{\text{th}}$  zero padded sub-filter given by:

$$h_k(n) = h(n + (k-1)M) \quad n=0, \dots, \frac{L-1}{2}, \quad k=1, \dots, R$$

$$= 0 \quad n = \frac{L+1}{2}, \dots, L, \quad k=1, \dots, R \quad (3)$$

Using simple properties of linearity only one DFT-IDFT pair of this length is required as all of the frequency domain sub-convolutions may be calculated then summed in the frequency domain then transformed back into the time domain. The resulting architecture is illustrated in Figure 4. This system performs convolution at a rate of  $1/M$  that of the sample rate of  $x(n)$ . It is clear that the length of the DFT-IDFT pairs may be chosen with rate reduction as the principal design criterion independent of FIR filter length. This simple architecture then allows relatively short DFT-IDFT lengths to be used to reduce the processing rate of arbitrarily high order FIR filtering or correlation operations, yielding overall simple designs.

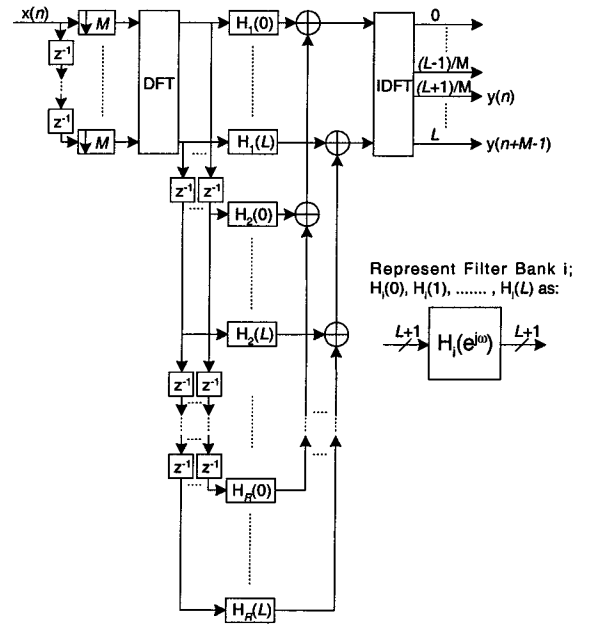


Figure 4. Parallel Sub-convolution Filter Bank Architecture

#### IV. Frequency Domain Fast LMS Architecture Employing Sub-Convolution and Sub-Correlation

Figure 5 illustrates a new frequency domain LMS architecture employing sub-convolution and sub-correlation. The architecture is similar to a traditional frequency domain LMS with block length 32 and 16 input sample overlap implementing 16 equalizer coefficients. However, in this architecture only 8-point FFT-IFFT pairs are required and it is extendable to 32 (or larger) coefficient length.

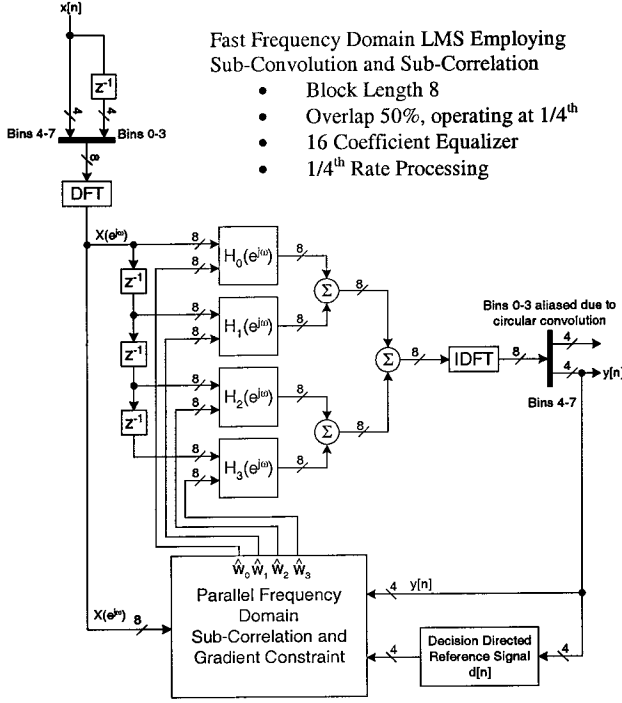


Figure 5. Parallel Frequency Domain Fast LMS Architecture Employing Sub-Convolution and Sub-Correlation

Assuming constant  $\alpha$  for all frequency bins, the architecture of Figure 5 performs the same as the traditional fast frequency domain LMS with block length 32 (32-point FFT-IFFTs) with 50% input block overlap, and 16 coefficients. Obviously in the 32-point FFT-IFFT case, if a frequency dependent step size  $\alpha$  is desired, there is more frequency resolution possible. All the mathematical operations of the 32-point fast LMS equalizer employing 32-Point FFT-IFFTs; as in Figure 2 with modifications such as 32 sample block inputs sliding 16 samples per clock and 32-point FFT-IFFTs, are performed in the new architecture illustrated in Figures 5 and 6. The convolution and correlation operations of the fast LMS have been replaced by sub-convolution and sub-correlation. There are other minor modifications that follow directly from linear and multirate systems theory. Figure 7 illustrates the function of the vector downsampler used throughout the architectures in Figures 5 and 6. The architecture of Figures 5 and 6 may be extended to arbitrarily long tap lengths, while still using 8-point FFT-IFFT pairs for the  $1/4^{\text{th}}$  processing rate reduction.

It should be noted that we have shown the frequency domain coefficients of Figure 5 to be updated every 4 system clock cycles; this corresponds to 16 time domain sample periods ( $T_s$ ). There may be advantages to updating more often, particularly with a very large number of coefficients. In the sub-convolution/sub-correlation architecture it is possible to update the coefficients more often than the traditional fast LMS frequency domain equalizer using an equivalent number of coefficients.

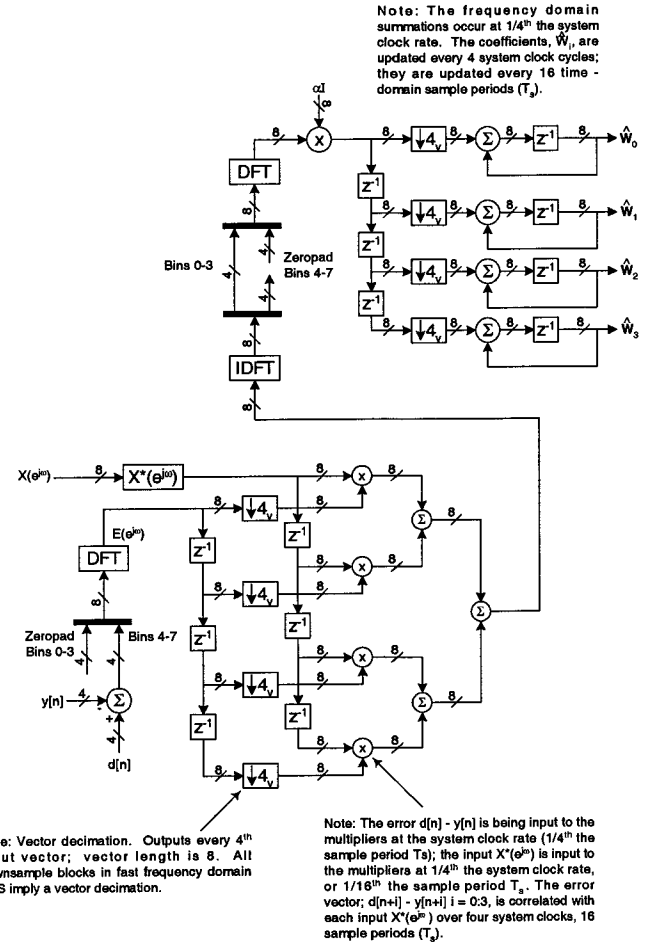


Figure 6. Parallel Frequency Domain Sub-Correlation and Gradient Constraint

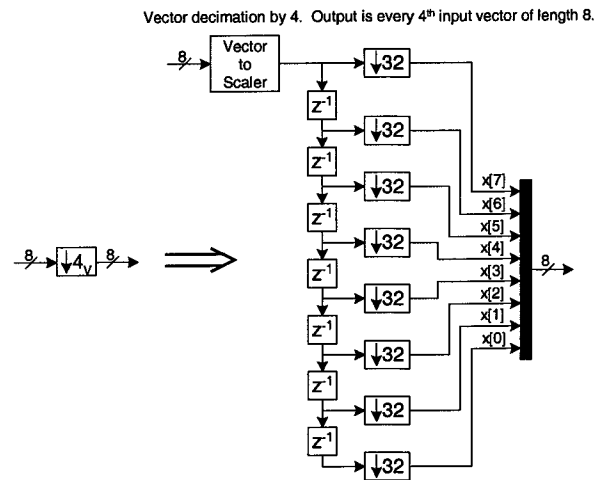


Figure 7. Length-8 Vector Downsample by 4

## V. Complexity Comparison

Here we use the number of complex multiplies as the basis for complexity comparison of the fast frequency domain LMS, time-domain block LMS, and fast LMS employing sub-correlation and sub-convolution. Further we assume a 16-tap equalizer and that the input  $x(n)$ , and filter coefficients,  $h(n)$ , are complex. When  $M$  is the DFT-IDFT length and  $M = 2(L+1) = 2^v$ , the number of multipliers

required by the FFT is given by  $\mu(M) = \frac{M(v-1)}{2}$  [3].

Note that there are numerous types of FFT-IFFT algorithms that could be used for comparison. We choose the radix two algorithm for convenience. From [1] the number of complex multiplies required by the block time-domain LMS is  $2(L+1)^2$ . The number of complex multiplies per equalizer output sample is then:

$$\frac{2(L+1)^2}{L+1} = \frac{2(16)^2}{16} = 32. \quad (4)$$

The number of complex multiplications per equalizer output of the fast 32-point ( $M=32$ ) frequency domain LMS is then:

$$\frac{\frac{5M(v-1)}{2} + 2M}{L+1} = \frac{\frac{5(32 \times 4)}{2} + 2 \times 32}{16} = 24. \quad (5)$$

The number of complex multiplications per equalizer output of the 8-point ( $M=8$ ) frequency domain LMS employing sub-convolution and sub-correlation is then:

$$\frac{\frac{5M(v-1)}{2} + 2 \times (4 \times M)}{M/2} = \frac{5 \times 8 + 2 \times 4 \times 8}{4} = 26. \quad (6)$$

Figure 8 is a plot of the approximate number of multipliers required to implement the three 16-coefficient architectures without hardware reuse. The fast LMS and block time-domain algorithms operate at  $1/16^{\text{th}}$  the input sample rate while the fast LMS using sub-convolution/correlation operates at  $1/4^{\text{th}}$  the input sample rate.

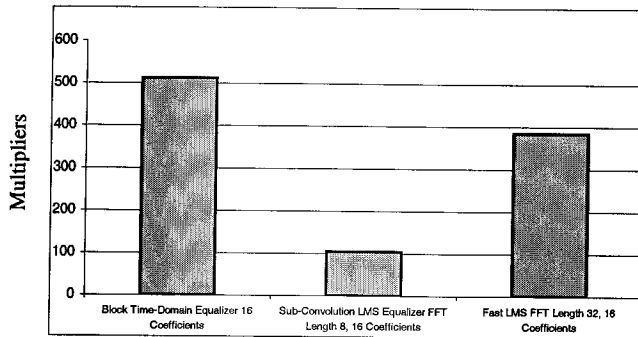


Figure 8. Number of Hardware Multipliers for Concurrent Implementation

## VI. Conclusion

We have presented a new fast LMS frequency domain equalizer architecture. The new architecture employs simple design techniques for designing parallel filter bank architectures based on the concept of separating a convolution into what we call sub-convolutions. These techniques allow arbitrarily long convolution or correlation to be performed using the overlap-and-save method with virtually any FFT-IFFT length. We have shown that the lower bound on FFT-IFFT lengths in the fast LMS algorithms created by tap length is removed using sub-convolution and sub-correlation techniques.

The non-fractionally spaced equalizer architecture presented operates at  $1/4^{\text{th}}$  the input symbol rate, uses 8-point FFT-IFFTs, has 16 coefficients and is extendable to higher tap lengths. Finally, the computation reduction over the serial implementation of time-domain block equalizer was demonstrated and the low complexity VLSI architecture was contrasted to the traditional frequency domain LMS equalizer whose FFT-IFFT lengths have traditionally been determined by tap length in concurrent hardware implementations. It was demonstrated with the architecture currently being developed, that the sub-convolution filter bank design technique is very useful for making tradeoffs between design complexity, computational efficiency, and processing rate.

## References

- [1] S. Haykin, *Adaptive Filter Theory*, Prentice Hall, Upper Saddle River, New Jersey 07458
- [2] John Proakis, *Digital Communications*, McGraw-Hill, Inc. New York, 1995.
- [3] A.V. Oppenheim, R. W. Schaffer, *Discrete-Time Signal Processing*, Prentice-Hall, Englewood Cliffs, New Jersey 1989.
- [4] W. Schober, F. Lansing, K. Wilson: JPL, E. Webb: NASA/GSFC, "NASA High Rate Instrument Study," National Aeronautics and Space Administration, Jet Propulsion Laboratory, JPL Publication 99-4, Jan., 1999
- [5] R. Sadr, P.P. Vaidyanathan, D. Raphaeli, S. Hinedi, "Parallel Digital Modem Using Multirate Filter Banks," JPL Publication 94-20, Jet Propulsion Laboratory, Pasadena, CA, August 1994.
- [6] Andrew A. Gray, "Very Large Scale Integration Architectures for Nyquist-Rate Digital Communication Receivers", PhD Dissertation, University of Southern California, Los Angeles, CA, May 2000
- [7] Gerald Grebowski, Andrew A. Gray, Meera Srinivasan, "Method and Apparatus for High Data Rate Demodulation", US Patent 6,177,835, Jan. 23, 2001
- [8] Andrew A. Gray, Meera Srinivasan, Marvin Simon, Tsun-Yee Yan, "Flexible All-Digital Receiver for Bandwidth Efficient Modulation", Int. Telemetry Conference, Nov. 1999
- [9] Meera Srinivasan, Chien-Chung Chen (JPL/Caltech), Andrew A. Gray, Gerry Grebowski (NASA/GSFC), "All-Digital High Data Rate Parallel Receiver," International Conference on Signal Processing Applications and Technology, Sept., 1998.
- [10] John J. Shynk, "Frequency-Domain and Multirate Adaptive Filtering," IEEE Signal Processing Magazine, January 1992.
- [11] Theodore Rappaport, *Wireless Communications*, PTR Prentice Hall, Englewood Cliffs, NJ, 1996.